Real Time Tracking Method by Using Color Markers

Jiamin Liu, Shuo Chen, Hongxing Sun, Yongxu Qin, Xibo Wang
School of Information Science and Engineering
Shenyang University of Technology
Shenyang, China
Christina.liu48@gmail.com

***Abstract*—To meet the user demands and to enhance integrating of markers into the environment that is close to nature as much as possible, this paper proposes a color marker-based method for tracking the target object in real time and registering a virtual three-dimensional (3D) object. Firstly, the contours of the color marker patterns on the video images are extracted by using adaptive threshold. Each marker pattern on the image is corrected and recognized by the template pattern. Secondly, the specified color markers are identified by using the $L_1L_2L_3$ and rgb models. Finally, the virtual objects are registered in real-time on the marker image by using the camera parameters derived from data of the color markers. The experimental results show that the virtual objects can be accurately registered on the markers of the video images. Compared to the ARToolKit software, this method is able to work in the environment with various illumination conditions.**

Keywords-Augmented reality; Tracking; Registration; Matching; Homography

I. INTRODUCTION

Augmented reality (AR) is a technology which supplements the real world with the virtual image, text and other information aligned with real scenes to augment perception and experiences for the real environment. It uses several disciplines, such as computer vision, computer graphics, human-computer interactive, display and so on. AR technology in the recent years has been widely used in various fields, such as industry, military, education, entertainment and medicine [1,2].

Nowadays tracking becomes the most popular topic for AR research because it is one of fundamental enabling technologies for AR. The version-based methods for real time tracking are divided into two categories, marker-based tracking and markerless-based tracking. The markerless tracking method registers virtual objects on the physical objects in the real scene according to a camera pose which is calculated by using the extracted data from nature features in the real scene. If nature features of the real target object are complex, the features are difficultly extracted and recognized. This results in a great amount of computational time for the process of feature matching. Therefore few applications developed by using markerless-based methods are presented. The marker-based method registers virtual objects on the specified artificial marker placed in the real scene. The artificial marker can be easily extracted and recognized because of obvious features. Thus it helps to correctly obtain camera pose and quickly track virtual objects on the known marker. The marker-based tracking has the advantages of accuracy and real-time tracking and robustness [3,4]. Therefore, the marker-based methods are widely applied to AR applications.

In the marker-based AR system, the black-and-white marker is generally used in the form of a pattern within a black square. The high contrast between black and white of the marker makes the marker's contour easy to be extracted. However, the adaptability for illumination and environment is weakness because a fixed threshold is used to recognize the binarization image with the marker. Furthermore, the black and white marker is not naturally blended with its surroundings. These disadvantages limit applications to some extent. In contrast, the color marker matches surroundings naturally. The published methods recognize the color marker by using the RGB and HSV models [5-7]. However, the two models are not flexible for illumination circumstances and cannot distinguish colors if their hues are close. Thus they are not able to feasibly handle all real video images. This paper proposes a method for tracking and registration that uses color markers with different hues and more close to colors in real scenes. Due to the limitation of the fixed threshold for color markers, an adaptive threshold for different colors is designed to extract the contour of color markers in the video image. It works well for various color markers and enhances the adaptability for varying illumination and environment. A new method for recognizing certain color marker in the image is presented. The method can exactly recognize color markers under the conditions of changeable illumination and environment. Then the virtual object can be accurately registered on the real scene by tracking these color markers in real time. Considering this method that has anti-interference and adaptability for natural environment, it can be further extended to other version-based tracking methods..

II. COLOR MARKER RECOGNITION

The triangular marker is more efficient than the quadrilateral one based on our previous work [8]. Therefore, the marker's pattern in this research is still an equilateral triangle with circles inside, and its color is of variety of similar and different hues. The multiple color markers are

978-0-7695-5150-0/13 $26.00 © 2013 IEEE
DOI 10.1109/ICVRV.2013.25

placed in a real scene at the same time to realize tracking and registration.

A. Marker Contour Detection

In order to recognize the marker's contour, the image of real world scene captured by a movable camera, i.e. target image, is first binarized. The conventional method uses the fixed threshold to handle the binarization image of black and white marker. However, to the image with various color markers, the fixed threshold method results in color lost, and is sensitive to illumination intensity. Therefore, an adaptive threshold for extracting color markers in the target image is designed in our method.

During binarizing the grayscale image, each pixel on the image has its own threshold which is obtained by first calculating the weighted average value of all pixels' gray values in an m×m region surrounding this pixel and then subtracting a constant from this value. Not only does the adaptive threshold improve the adaptability to the complex background and light intensity, but also avoids the loss of the markers' contours in the binary image. Through the experimental test, the optimal result can be obtained by using the m values of 41 and constant of 5.0.

Each pixel in the binary image is marked as either an edge pixels or a non-edge pixel. The edge pixels of the marker are detected by using Canny Operator. Based on the Douglas-Peucker algorithm [9,10], the detected pixels are extracted from the feature points detected to form a polygon. The polygon is checked whether it is a triangle. The triangles that are larger or smaller than the area of the given triangle are discarded. Thus the computational time for the subsequent template matching is reduced.

B. Image Correction

Due to the different camera views, the triangular marker on the image may be distorted to form a non-equilateral triangle. Thus the triangle must be corrected and normalized by affine transform of rotating, stretching and scaling. This ensures that size of the normalized triangular region is the same as the template image.

Transformation is completed by calculating a perspective matrix, which uses four pairs of corresponding points between the target image and original template'. The first three pairs consist of the vertices of triangles on template and target images. The fourth pair is the center points of the triangles on the template and target images [8]. These four pairs are used to calculate the perspective transformation matrix to complete the image correction.

C. Template Image Matching

After the image correction, the target image needs to match with the template to determine the marker pattern. The corrected image must be converted into the gray image before matching, and then matched to the template via an algorithm of correlation coefficient [11,12]. The algorithm uses the relative values of the average gray values of the target image to match with ones of original template image by using (1).

$$R(x,y)=\frac{\sum_{x}^{M}\sum_{y}^{N}\left[S(x,y)-\bar{S}\right]\left[T(x,y)-\bar{T}\right]}{\sqrt{\sum_{x}^{M}\sum_{y}^{N}\left[S(x,y)-\bar{S}\right]^2\sum_{x}^{M}\sum_{y}^{N}\left[T(x,y)-\bar{T}\right]^2}} \tag{1}$$

where M and N are the number of pixels in rows and columns in the image respectively. S is the average value of template image; T is the average value of target image; R(x, y) is the match value between pixels (x, y) of two images. It can reduce the impact of light change on the image. Its value ranges from 0 to 1. The closer is R to 1, the more matching is image to template.

D. Color Matching

The gray values of different color markers can change under different light intensities. If the gray values of two colors in the captured image are similar to that of one template after the color images becomes grayscale, this results in image mismatch. Therefore, it is impossible to use only one template to distinguish different colors of the markers.

To identify the specified marker on the target image, the two color models are exploited to match the colors of two images. The first model is called $L_1L_2L_3$ model [13,14]. The L_1, L_2 and L_3 values of this model for each pixel on the image are respectively calculated from the RGB value of the pixel by using (2).

$$\begin{cases} L_1 = \dfrac{(R-G)^2}{(R-B)^2+(R-G)^2+(G-B)^2} \\ L_2 = \dfrac{(R-B)^2}{(R-B)^2+(R-G)^2+(G-B)^2} \\ L_3 = \dfrac{(G-B)^2}{(R-B)^2+(R-G)^2+(G-B)^2} \end{cases} \tag{2}$$

The $L_1L_2L_3$ model is more suitable for the scenarios where the light intensity and direction of view point change often. In most cases, the colors between two images can be successfully matched by using this model. However, there is one exception where the hue of a color marker is complementary hue of another marker. The L_1, L_2 and L_3 values of the colors on the two markers calculated by the model are identical. This leads to the image mismatch. Therefore, the rgb model is used for further matching. The r, g and b values of the model are respectively calculated by using (3).

$$\begin{cases} r(R,G,B)=\dfrac{R}{R+G+B} \\ g(R,G,B)=\dfrac{G}{R+G+B} \\ b(R,G,B)=\dfrac{B}{R+G+B} \end{cases} \tag{3}$$

The rgb model avoids the color mismatch occurred in the $L_1L_2L_3$ model. However, the rgb model cannot distinguish the markers with similar hue. Therefore, the two color models are used during the color matching.

To recognize the marker's color, the color values of all pixels within the triangles of the template and target images are re-calculated according to the above color models. This process takes amount of computational time. To improve computational performance, only a number of pixels within the triangles of the template and target images are picked up. Then the color match value R(x, y) of the selected pixels is calculated with using (4).

$$R(x,y)=\frac{\sum_{(x,y)}(S(x,y)-T(x,y))^2}{\sqrt{(\sum_{(x,y)}S(x,y)^2)^2+(\sum_{(x,y)}T(x,y)^2)^2}} \tag{4}$$

If the R value is close to 0, the image is accurately matched.

III. TRACKING AND REGISTERING

In order to track and register virtual objects in real time on the markers of the video images, it is necessary to obtain camera pose in the world coordinate system. This process is called camera calibration. The goal of camera calibration is to construct the corresponding relationship between the key points on the maker in the real scene with the points on the target image [15]. The process is described by using (5).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda M[r_1 \quad r_2 \quad r_3 \quad t]\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5}$$

where (x, y, z) are the coordinates of the point on the marker in the world coordinate; (u, v) is the mapping point in the image; M and [r1 r2 r3 t] are respectively camera intrinsic and external parameters; [r1 r2 r3] and [t] are respectively a 3-by-3 rotation matrix and a 3-by-1 translation vector which describe the position of the object relative to the camera coordinate system in terms of a rotation and a translation; λ is a scale factor. The camera intrinsic parameters are constants during the image processing because they are the same from all views. Therefore, the key issue is that the camera external parameters must be determined.

Assume that each marker is on own independent coordinate system. The camera parameters are computed by a homography H which is one-to-one projective mapping between the marker coordinate system and image coordinate, where is a 3-by-3 matrix, $H=[h_1,h_2,h_3]$.

The homography H that maps a planar object's points onto the imager is

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = [h_1 \quad h_2 \quad h_3]\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{6}$$

According to (5), the matrix H is

$$H=[h_1 \quad h_2 \quad h_3]=\lambda M[r_1 \quad r_2 \quad r_3 \quad t] \tag{7}$$

H uses multiple images of the same object to compute both the individual translations and rotations for each view as well as the intrinsic. Rotation is defined by three angles and translation is defined by three offsets. Hence there are six unknown values for each view. H can be figured out by using four pairs of given points between the marker and image planes. Equation (7) is transformed into (8)

$$\begin{cases} r_1=\lambda M^{-1}h_1 \\ r_2=\lambda M^{-1}h_2 \\ r_3=r_1\times r_2 \\ t=\lambda M^{-1}h_3 \\ \lambda=1/\|M^{-1}h_1\| \end{cases} \tag{8}$$

Therefore, the camera extrinsic parameters are obtained according to the above equations. When a virtual object is registered, we use OpenGL API to draw the virtual object according to the camera parameters, because the process of OpenGL generate target scene is similar to the camera imaging process.

IV. EXPERIMENTS AND COMPARISON

The method is implemented in Visual studio 2008 and OpenGL library under Windows XP. The experiments are carried on a computer with Intel Centrino Duo CPU 2.2 GHz and RAM 2 GB. Camera resolution is 640×480. A virtual teapot rendered by OpenGL is registered on each marker in the real scene.

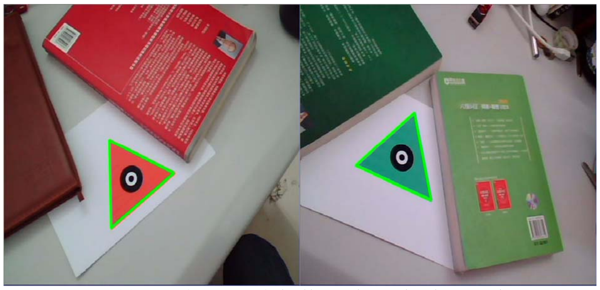
This method is tested under the conditions of multiple color markers with different and similar hues, various illumination and complex backgrounds. The experimental results are compared with those obtained by the ARToolKit.

A. *Recognizing Color Markers*

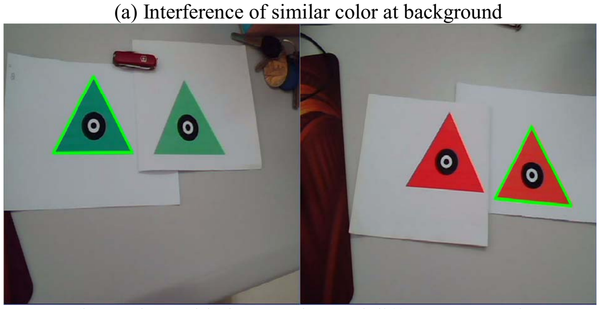
Firstly, the homo-color markers are placed next to an object having similar color to the marker's (Fig. 1(a)), and next to another markers having the same hue and different saturations with the markers (Fig. 1(b)). It shows that the method can recognize the markers. The recognized markers in Fig. 1 are highlighted by the green triangles. Next, the multi-color markers are placed along with the homo-color markers in the scene (Fig. 1(c)). Although the multi-color markers are interfered with the homo-color markers because they have the same color, they can also be recognized. Finally, the four homo-color markers are all placed in the scene (Fig. 1(d)). As a result, they have been completely recognized at the same time.

B. *Tracking and Registering*

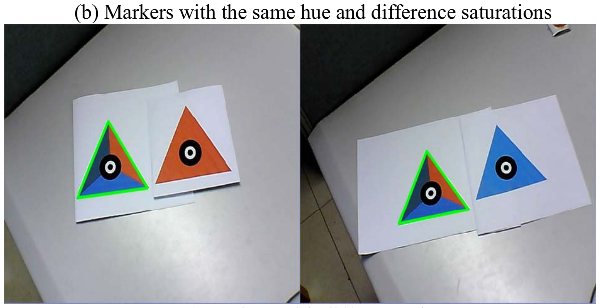
1) Marker on single plane: Four markers with different colors are placed on the same plane for tracking and registering. These markers can be real-time tracked and the four virtual teapots can be accurately registered on these markers, shown in Fig. 2.



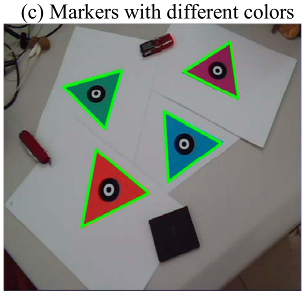
(a) Interference of similar color at background



(b) Markers with the same hue and difference saturations



(c) Markers with different colors



(d) Multiple color markers recognition

Figure 1. Marker recognition

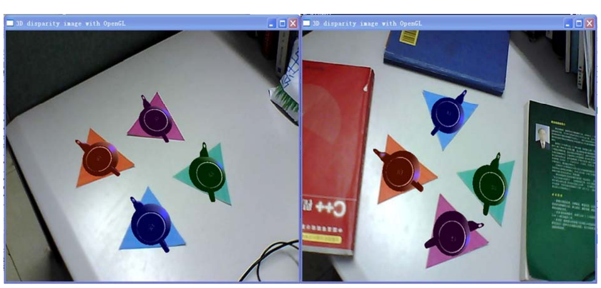


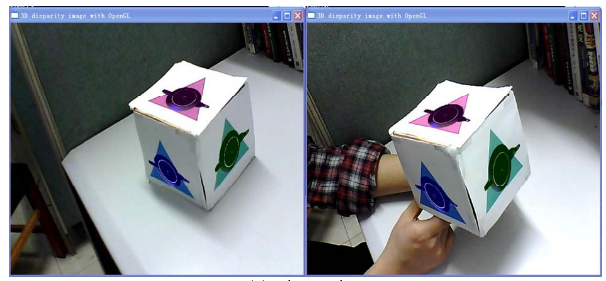
Figure 2. Registration in the same plane.

2) Markers on the multiple planes: Four markers with different colors are put on the four faces of a cube, shown in Fig. 3. Through shifting the camera positions or rotating the cube, the markers on the cube can be real time tracked and at least two virtual teapots are able to be registered on three visible faces of the cube each time.

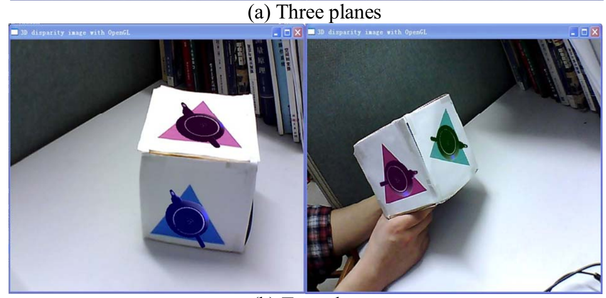
C. Varying Illumination

We set up four circumstances in which natural illumination varies from strong to weak under daylight. Tracking and registering by this method and the ARToolKit software were tested under these four circumstances. The ARToolKit uses a black square marker with a code inside and register a virtual cube on the marker.

As shown in Fig. 4, our method can complete the registration under all circumstances, but the ARToolKit can only do under the two circumstances (Fig. 4(b) and (d)). The results show that our method has stronger adaptability for varying illumination than the ARToolKit.



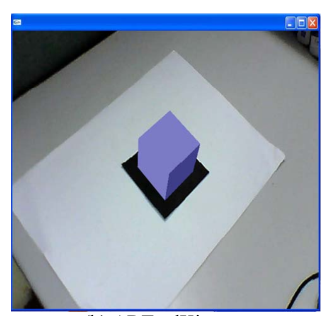
(a) Three planes



(b) Two planes

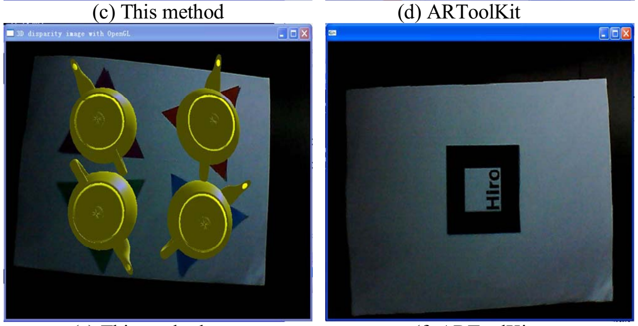
Figure 3. Registration on the different planes.

(a) This method

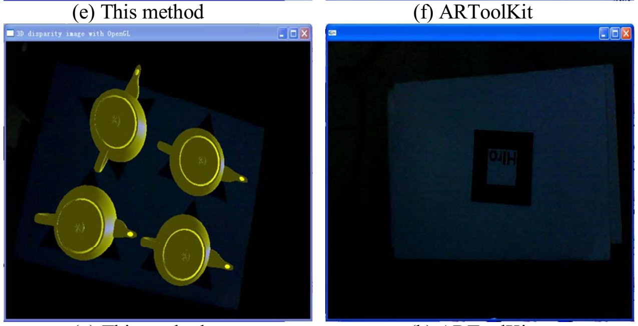


(b) ARToolKit

(c) This method (d) ARToolKit



(e) This method (f) ARToolKit



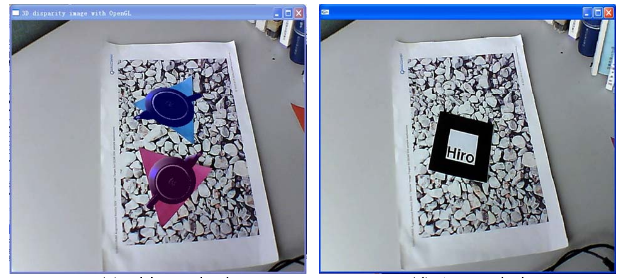
(g) This method (h) ARToolKit

Figure 4. Results by two methods under varying illumination.

D. Complex Backgrounds

To further test the performance of two methods, the markers of two types were respectively placed under the complex background. It is remarkable that this method is still able to track and register on the markers, but the ARToolKit is not able to due to the interference of background, shown in Fig. 5.

(a) This method (b) ARToolKit



(c) This method (d) ARToolKit

Figure 5. Under the backgrounds of stone texture.

TABLE I. EXECUTION TIME OF 15 FRAMES

Frames	Time/ms
10	6.692
20	6.655
30	6.529
40	6.582
50	6.651
60	6.586
70	6.607
80	6.480
90	6.758
100	6.640
110	6.601
120	6.842
130	6.639
140	6.593
150	6.507

E. Execution Time

The method has been tested based on four markers with different colors. 150 video frames are collected. The execution time of 15 frames in the video is listed in Table I. The average running time is 6.624ms by this method. The average running time of ARToolKit is 5.743ms[8]. Although the running time is slightly longer than the ARToolKit's, it can meet the requirement of real-time.

V. CONCLUSIONS

A novel method in this paper is proposed based on color markers. This method can identify the color markers by using the proposed color model in order to real-time track the marker and register virtual object on the marker in the real scene. The experimental results show that the method has good accuracy and robustness, strong anti-interference, strong adaptability of illumination.

For this reason, the marker is not limited to any color. Any color requested by users can be used on the marker. This not only riches the color information, but also increases the number of markers. Moreover, the marker is efficiently integrated into a real scene.

However, a few color markers in this method are simultaneously used to compute the camera positions and orientations. This results in a slightly long computational time even though the method can real-time track the color markers and register virtual objects. When the marker's size in captured image is small, this method can't track the marker accurately. Therefore, the method needs to be further improved.

ACKNOWLEDGMENT

The authors would like to thank the participants and the anonymous reviewers for their comments and feedback. This research was supported by the Science and Technology Department of Liaoning Province in China (Grant No. 201102164) and by Liaoning BaiQianWan Talents Program (Grant No: 2012921041).

REFERENCES

[1] Zhou F, Duh H B L, Billinghurst M, "Trends in augmented reality tracking, interaction and display: A review often years of ISMAR", Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society, 2008, pp.193-202.)

[2] Van Krevelen D W F, Poelman R, "A survey of augmented reality technologies, applications and limitations", The International Journal of Virtual Reality,vol.9 2010, pp.1-20.

[3] SHA Jun-yuan, QIN Wen-hu, ZHAO Zhen-xu, "Study of augmented reality based on artificial", Computer Engineering and Design, 2008,29(18), pp.4819-4821.

[4] Park Hanhoon, Park Jong, "Invisible marker-based augme-nted reality", International Journal of Human-Computer Inter-action, vol.26, 2010, pp.829-848.

[5] Li Yu, WANG Yong-tian, LIU Yue, "Study of color fiducial based registration algorithm for augmented reality", Journal of System Simulation,vol.20, 2008, pp.654-661.

[6] XU Tong, WANG Yong-tian, "Study of three-color fiducial registration for augment reality", OPTICAL TECHNIQUE, vol.26, 2000 , pp.535-540.

[7] Belghit H, Zenati-Henda N, Bellabi A, "Tracking colormar-ker using projective transformation for augmented reality application", Multimedia Computing and Systems (ICMCS), 2012 Int-ernational Conference on IEEE, 2012, pp.372-377.

[8] LIU Jia-min, AN Le-xiang, CHANG Yan, ZHU Shitie, YANG Dan, "Three-dimensional registration method based on triangular marker in augmented reality", JOURNAL OF SHENYANG UNIVERSITY OF TECHNOLOGY,2013, pp.78-84.

[9] Sebastian Lieberknecht, Selim Benhimane, Peter Meier, Nassir Na-vab, "Benchmarking template-based tracking algorithms", Virtual Reali-ty, 2011, pp.99-108.

[10] Liang Lin, Wangyong Tian, Yue Liu, Caiming Xiong; Kun Zeng, "Markerless registration based on template tracking for augmented reality". Multimedia Tools and Applications,vol.41, 2009, pp.235-252.

[11] Li Zhuo, QIU Hui-juan, "Fast image matching based on correlation coefficient" Transactiong of Beijing Institute of Technology,vol.27, 2007, pp. 998-1000.

[12] WEI Ji-hua,FU Si-hua, "Real-time matching algorithm based on the property of correlation coefficients", OPTICAL TECHNIQUE,vol.34, 2008, pp.100-102.

[13] Torralba A, Fergus R, Weiss Y, "Small codes and large image databases for recognition", Computer Vision and Pattern Recognition, IEEE Conference on IEEE, 2008, pp.1-8.

[14] Gevers T, Smeulders W M, "Color based object recognition", pattern recognition,vol.32, 1999, pp.453-464.

[15] Jiang Qinyun,Wang Cheng,Li Lijun,Guan Tao, "Study on registration algorithm in augmented reality", Computer & Digital Enginee-ring,vol.34, 2006, pp.38-40.